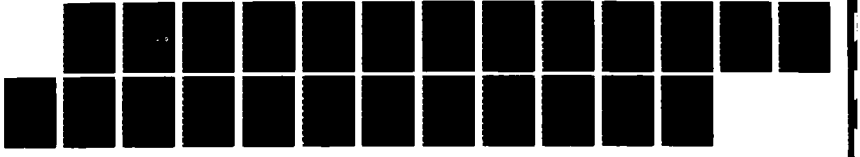
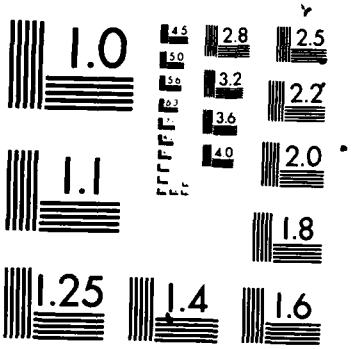
AD-A172 593 DETERMINING THE EXISTENCE OF BIMODAL DISTRIBUTIONS IN THE DATA FROM STATI (U) DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER ANN... G FOWLER SEP 86 1/1

UNCLASSIFIED DTNSRDC/SME-86/43 F/G 12/1 NL





David W. Taylor Naval Ship Research and Development Center
Bethesda, MD 20084-5000

DTNSRDC/SME-86/43 September 1986

SHIP MATERIALS ENGINEERING DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

DETERMINING THE EXISTENCE OF BIMODAL DISTRIBUTIONS IN THE DATA FROM STATIC PANEL IMMERSION TESTING

by
Gary Fowler
Mathematics Department
USNA

Determining the Existence of Bimodal Distributions in the Data from Static Panel Immersion Testing

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A172593

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) DTNSRDC SME-86/43	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION David Taylor Naval Ship R&D Center 6b OFFICE SYMBOL (If applicable) 2844	7a NAME OF MONITORING ORGANIZATION David Taylor Naval Ship R&D Center Propulsion Aux. Systems Dept., Code 2759
6c ADDRESS (City, State, and ZIP Code) Bethesda, Maryland 20084-5000	7b ADDRESS (City, State, and ZIP Code) Bethesda, Maryland 20084-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Technology 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 64710N PROJECT NO: TASK NO: R0371012 WORK UNIT ACCESSION NO: 12759446

11 TITLE (Include Security Classification)
Determining the Existence of Bimodal Distributions in the Data From Static Panel Immersion Testing.

12 PERSONAL AUTHOR(S)
Gary Fowler

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM 6/84 TO 9/86	September 1986	20

16 SUPPLEMENTARY NOTATION

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Testing; Prediction; Service Life; Antifouling Coatings, Static Panel Immersion.

19 ABSTRACT (Continue on reverse if necessary and identify by block number)
An algorithm is presented that aids in deciding whether a sample is from a single population or a mixture of two populations. It is a combination of two established algorithms, the EM algorithm and the minimization of AIC. When tested on simulated data the algorithm performed well.

20 DISTRIBUTION AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Vincent J. Castelli	22b TELEPHONE (Include Area Code) 301-267-2853 22c OFFICE SYMBOL 2844

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

TABLE OF CONTENTS

LIST OF TABLES

Abstract

An algorithm is presented that aids in deciding whether a sample is from a single population or a mixture of two populations. It is a combination of two established algorithms, the EM algorithm and the minimization of AIC. When tested on simulated data the algorithm performed well.

Description of Problem

Research into the development of improved antifouling shipbottom coatings makes extensive use of static panel immersion tests to screen developmental materials. A currently used procedure is to expose twelve to twenty 10" by 12" test panels coated with the developmental coating at one or more of the Navy's exposure sites (Miami, FL, Half Moon Bay, CA and Nawiliwili, HI). These panels are evaluated on a quarterly basis to estimate the amount of fouling as a measure of the effectiveness of the coating. These life experiments can require very long periods of time to complete, usually three or more years. The quarterly evaluations are reported to DTNSRDC, where it is desirable to make judgments regarding the progress of the experiment. This report suggests a method of analysis of the interim data, which would aid in judging the progress of the experiment. It has been observed by Becka [1983] that the fouling times for some samples of antifouling coatings exhibit a bimodal distribution, i.e. a distribution with two local maxima. It is believed that this can be explained by viewing the sample as coming from two different populations, rather than the usual view that the sample comes from a single population. For example, a sample of twelve panels may consist of five panels fouling at a different rate than the other seven panels. Of course, at the beginning of the experiment it was believed that all panels were identical. Thus they should represent a

single population. It is only after the experiment has progressed for some time that two clusters of panels may become apparent, one cluster having substantially more fouling than the other. The problem to be analyzed in this report is twofold. First, is there sufficient evidence to believe that the sample has items drawn from two populations or just one? Second, how many and which panels belong to the two populations?

Bimodal Distributions

It will be assumed that the data consists of a sample of M values of the variable FL, where FL is the value report from the exposure site. It is generally a percentage, between 100 and 60, rating the amount that the panel is not fouled. Generally conclude that if the FL's are a sample from two populations, then the largest K are from one population and the remaining (smallest) M-K are from the other population; K is to be determined and may be zero or M. Especially when K is small, e.g. 10% of M, one should be careful to not conclude that there are two populations without further investigation.

Several phenomena complicate the investigation. Among these is the fact that random samples can exhibit quite large variations. It is possible that there will be outliers even when there is only one population. An outlier is an observation at a great distance from the expected fouling level. The statistical modeling of the fouling process can be very sensitive to extreme values. If the outlier is truly from the population being studied, then it is important to include it, since it contains information not included in the remainder of the sample. Hence, it is important to physically examine outliers

to determine if they are in fact special or simply a manifestation of the randomness of the sampling process. On the other hand, models that employ estimates assuming that the data is a sample from a single population should be used with care when the sample appears to have come from two populations.

Further complicating the analysis are masking and swapping. These terms are used to describe the problems arising from the fact that there is a gray area between two populations. As an illustration consider the histogram of fouling levels FL in Figure 1.

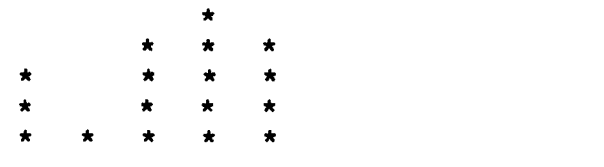


Figure 1

The data point at 95 is masked by the data points at 100, in that they cause this point to appear to be part of the leftmost cluster. Of course, it is masked, perhaps more strongly, by the data points at 90, 85 and 80. It is also possible that one of the points at 100 belongs to the cluster on the right rather than on the left. It has been swapped. It is unlikely that any statistical analysis will completely sort out these kinds of problems. It can however call them to the investigator's attention.

A common technique for modeling samples from two populations is to use a mixture of two distributions. Suppose f(x;P) is the probability function describing one of the two populations, where P is a vector of parameters and x is the value of the variable, in this setting FL; and g(x;Q) is the probability function describing the other population. The mixing proportion, MixProp,

is a number between 0 and 1 and

$$\text{MixProp}.f(x;P) + (1-\text{MixProp}).g(x;Q) \qquad \text{EQ. 1}$$

is the probability function describing a mixture of the two populations. Each item in a sample of size M from the mixture can be thought of as coming from the population f(x;P) with probability MixProp. Said another way, in a sample of size M one expects, on the average, for K=M.MixProp of the items in the sample to have come from the population f(x;P). Hence, to decide if there are two populations represented in a sample, it is sufficient to estimate the parameter MixProp.

As already noted the phenomenon under investigation was first observed by Becka [1983] as a bimodal distribution. She observed that the reported data had a histogram similar to the one in Figure 1. It is, of course, not necessarily the case that a mixture of two distributions is bimodal. Consider, for example, a mixture of two binomial distributions

$$\text{MixProp}.\text{bin}(x;p_0,N) + (1-\text{MixProp})\cdot\text{bin}(x;p_1,N) \qquad \text{EQ. 2}$$

where

$$\text{bin}(x;p,N) = (N!/[x!(N-x)!])p^x(1-p)^{n-x} \text{ for } x=0,\ldots,N \qquad \text{EQ. 3}$$

is the usual binomial density with N trials and probability of success p. It can be shown that if p_0=0.95, p_1=0.8 and N=20, then MixProp must be between 0.296 and 0.347 in order for the mixture in EQ. 2 to be bimodal. See Table 1 for the interval of MixProp that will give a bimodal distribution for various values of p_0 and p_1. Even though the mixing of two populations was first observed in the bimodal case, it is now apparent that a test for mixing rather than for bimodality is needed.

Table 1

INTERVAL FOR MIXING PROPORTION
THAT RESULTS IN A BIMODAL DISTRIBUTION

		p_o = 0.95	0.9	0.85
p_1	0.65	(.041,.98)	(.19,.656)	(.402,.434)
	0.7	(.10,.868)	(.315,.456)	never bimodal
	0.75	(.195,.547)	never bimodal	never bimodal
	0.8	(.296,.347)	never bimodal	never bimodal

Discussion of General Methodology

The statistical literature is replete with discussions and suggestions for determining both outliers and mixing proportions; for a survey see Beckman and Cook [1983]. Two methods standout for applications of the type required in this report. They are the EM algorithm, Dempster, et. al. [1977], and AIC minimization, Akaike [1977]. A general discussion of these methods as they apply to the present context follows. The EM algorithm will be discussed first, since it is needed to compute AIC.

The maximum likelihood principle is essential to both methods. It is the naive notion that given a collection choices one should choose the one that is most likely. In many common situations there is a closed form for the maximum likelihood estimator of the unknown parameter. There is not a closed form solution for the unknown parameter, MixProp, in our setting, however. The EM algorithm provides an iterative, numerical method for approximating the unknown parameters. For the model which is a mixture of two binomials, EQ. 2, and a sample $x_1,\ldots,x_m$.

$$p_j = (\sum x_i \text{bin}(x_i;\text{Old}p_j,N)/\text{mixbin}(x_i))/(NM) \quad j=1,2 \qquad \text{EQ. 4}$$

$$\text{MixProp} = \text{OldMixProp}\cdot(\sum(\text{bin}(x_i;p_o,N)/\text{mixbin}(x_i)))/M \qquad \text{EQ. 5}$$

where mixbin(x) is the mixture in EQ. 2 and both sums run from i=1 to M.

This algorithm necessarily converges. However, it may be slow and it may converge to a local extreme rather than the absolute maximum, if the starting point is not carefully selected. This later difficulty usually can be avoided by selecting several starting points and selecting the maximum that is the largest among those generated by the various starting points.

The EM algorithm can be applied in this setting as follows. Apply the algorithm for MixProp=K/M for K=1,...,M where M is the sample size. For each of these estimates compute the likelihood. Actually it is easier, and more common to compute the negative of the natural logarithm of the likelihood:

$$-\sum \ln(\text{mixbin}(x_i;p_o,p_1,\text{MixProp},N) \qquad \text{EQ. 6}$$

where the sum runs from i=1 to i=M and $\text{mixbin}(x_i;p_o,p_1,\text{MixProp},N)$ is the mixture of two binomials as in EQ. 2. Select the estimate of MixProp that yields the largest likelihood (smallest negative ln likelihood). When using this procedure the estimate of MixProp will usually not be of the form K/M. That is MixProp•M, which should be interpreted as the number of sample units from the population with parameter p_o will not necessarily be a whole number. For example, the statistical analysis may report that 5.7 of the sample units are from one population and the other 10.3 are from the other population. One is reminded that masking and swapping are present. When applied to simulated data this procedure works well, but can be improved by using the AIC minimization principle described next.

The problem of selecting the "best" model from several competing models is a common problem. Akaike [1977] suggested employing the principle of minimizing the negative entropy in the selection process. For a specific model define its AIC by

AIC = $-2 \cdot \ln$(maximum likelihood)

$+ 2 \cdot$(number of independently adjusted parameters). EQ. 6

To apply this principle in the present setting, select the model with the smallest AIC from among the models

MixProp.bin(x;p_o,N) + (1-MixProp)$\cdot$bin(x;p_1,N) EQ. 7

where MixProp=K/M indexes the models for K=1,...,M. That is, index the models under consideration by the number of sample units from the population with parameter p_o. Hence there are M models to select from. For K = 1,..,M-1 there are two independently adjusted parameters, p_o and p_1. For K=M there is only one parameter, namely P_o since all sample units are from one population.

This procedure differs from the procedure using the EM algorithm in two respects. The values of MixProp/M = K are restricted to being whole numbers. More significantly, observe that if it were not for the second term in the expression for AIC, namely $2 \cdot$ (number of independently adjusted parameters), then minimizing AIC is equivalent to maximizing the likelihood. This extra term in AIC results in a preference for selecting the model that says the data is from a single population. Akaike [1977] claims that in fact AIC corrects for a bias in the maximum likelihood principle that causes a model with fewer parameters to be rejected too often.

The algorithm based on AIC has been found to be sensitive to the estimate of the maximum likelihood. In particular, if the maximum likelihood estimates of the parameters p_o and p_1 not accurately made, then the results of minimizing AIC can be quite unsatisfactory. The values of AIC differ very little from one model to the next. There are no tables of the probability distribution of

AIC, hence is it difficult to judge whether small differences are significant or not. This difficulty is alleviated somewhat by computing good estimates of p_0 and p_1. The EM algorithm works well when applied to this problem. In this setting apply it as above, but do not use it to update the estimate of MixProp, since it is fixed for each model. Compute new estimates of p_0 and p_1 only.

Results of Testing the Algorithm

The algorithm described in the previous section was programed in Turbo Pascal and applied to four data sets. These data sets were generated so as to have certain properties: (1) p_0=0.95, p_1=0.7, MixProp=0.35, strikingly bimodal; (2) p_0=0.95, p_1=0.7, MixProp=0.75, less strikingly bimodal; (3) p_0=0.95, p_1=0.8, MixProp=0.29, not bimodal but having a relatively large variance; and (4) p_0=0.9, MixProp=1, a single population. Graphs of these are provided in Appendix B. Listings of the programs, with notes, are provided in Appendix A. One hundred sets of data were generated for each of these four cases. The combined AIC and EM algorithms were used to estimate K= MixProp*M, the number sample units from the population with parameter p_0. The results of this simulation are summarized in Table 2, which contains the counts of the number of times estK was the estimate of MixProp*M. Throughout this simulation M=16 and N=20.

Table 2

Frequency Table of Estimated M*MixProp

estK	(1)	(2)	(3)	(4)
1	1	0	2	0
2	4	0	6	0
3	6	1	10	0
4	6	0	8	2
5	22	3	*9	0

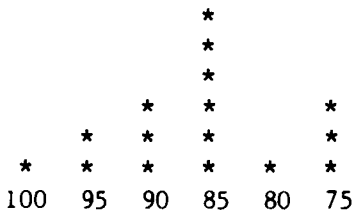
Table 2 - Continued

6	*16	1	3	0
7	15	0	7	0
8	12	8	7	0
9	12	8	6	1
10	4	10	3	0
11	1	19	1	0
12	0	*13	4	0
13	0	15	0	0
14	0	12	1	0
15	0	4	3	0
16	1	6	29	*97

(1) p_0=0.95, p_1=0.70, MixProp=0.35
(2) p_0T=0.95, p_1=0.70, MixProp=0.75
(3) p_0=0.95, p_1=0.80, MixProp=0.29
(4) p_0=0.90, MixProp=1.0
An * marks the value of K=M•MixProp.

If there were no variation in the simulated data sets, and if they came from the prescribed population with certainty, and if the algorithm worked perfectly, then the numbers preceded by an * in Table 2 would be 100. The algorithm works quite well in cases (1) and (2) and amazingly well in case (4). Case (3) and case (4) show the disposition of AIC to favor the model of a single population. Case (3) is generated from a mixture that is not bimodal, but has a large variance.

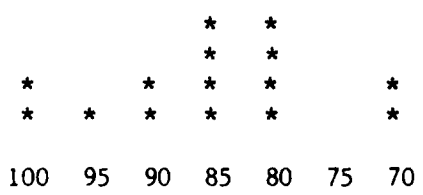
Consider some illustrations from case (3):



The parameters that were used to generated this data were p_0=0.95, p_1=0.8, and K=4.64. That is one expects the letmost 5 data points to be from one population and the rightmost 11 to be from another. On the other hand the estimated

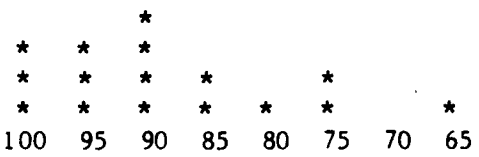
values are $estp_0$=0.859 and estK=16. That is the sample is from a single population with parameter 0.859. The problem here is that since these distributions are so close, there is a significant amount of swapping and masking, resulting in a distribution that appears to be that of a single population rather than that of a mixture.

Another illustration from this data set follows:



The estimated parameters this time are $estp_0$=0.999, $estp_1$=0.793, and estK=1. That is one of the data points at 100 is from one population and the remaining fifteen are from another. Of course, there would be no way of knowing which of the two test panels that are 100% unfoulded belong to the two different populations. One would either treat them both as coming from a population distinct from that of the other fourteen, or treat the entire sample of sixteen as coming from a single population.

Another illustration from this data set follows:



The estimated parameters are $estp_0$=0.935, $estp_1$=0.792, and estK=10. This data set was generated with K=5. That is the expected number of data points in the leftmost population is five, not the estimated ten. Although ten seems more reasonable, than five, neither reflect the possibility that the two points at 85 could belong to the leftmost group.

This algorithm works best when the two populations are widely separated or when there is only one population. This is not surprising since swapping and masking are less important when the populations are not close. When the populations are distinctly separated, then the mixture distributions tend to be bimodal. Hence, one observes that the algorithm works best separating the populations when the mixture is bimodal. The algorithms excellent performance in recognizing a single population can be attributed to the fact that AIC is adjusted in favor of selecting simpler models.

CONCLUSIONS

The proposed algorithm works well, but does not make the decision for the experimenter. This algorithm should be used as a decision aid and not as a decision rule. One should keep in mind the problems of swapping and masking and remember that the purpose of the analysis is to decide which panels might require further investigation or monitoring.

The algorithm described in this report has been tested only on simulated data. It should be tested on actual data that is well understood. That is, it should be tested on actual field data that apparently comes from a single population and on data that seems to come from two populations. Finally, it should be used as an aid in making interim judgments to determine if it is useful for that task.

REFERENCES

1. AITKIN, M and WILSON, G. T. (1980). Mixture models, outliers, and the EM algorithm. Technometrics, 22, 325-331.

2. AKAIKE, H., (1977). On entropy maximization principle. Proc. Symposium on Application of Statistics, ed. P.R. Krishnaiah, pp 27-47. Amsterdam: North-Holland.

3. BECKA, A. M., "Improved Analysis of Static Panel Immersion Testing Results," Journal of Coatings Technology, 55(1983) no. 703, 51-54.

4. BECKMAN, R. J., and COOK, R. D., (1983). Outlier.....s (with discussion). Technometrics, 25, 119-163.

5. DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion). J. Roy. Stat. Soc., B, 39, 1-38.

6. KITAGAWA, G. (1979). On the use of AIC for the detection of outliers. Technometrics, 21, 193-199.

APPENDIX A
PROGRAM LISTINGS

This appendix contains listings of the following program and procedures:

AICMIXBI.PAS - the main program;
BINTABLE.PAS - a procedure for computing binomial probabilities;
MIXTABLE.PAS - a procedure for computing probability mixtures;
NEGLLIKE.PAS - a procedure for computing negative log likelihoods
NEWP.PAS - a procedure for computing new values of p_0 and p_1.

```
program AICMIXBIN;
{Approximates the parameters of a mixture of two binomials via the AIC and EM
algorithms.}

Label
  DataError;

Const
  N = 20;
  ssize = 16;
  er = 0.0005;

Type
  Sample = array[1..ssize] of integer;
  ProbTable = array[0..N] of real;
  Out = record
        prob0: real;
        prob1: real;
        mixprob: real;
        DataNum: integer;
        end;

Var
  p0, p1, mixprop, oldp0, oldp1, newll, aic, minaic : real;
  indx, jndx, start, finis: integer;
  cf0, f0, cf1, f1, cm, m: ProbTable;
  data: Sample;
  DataFile: file of sample;
  OutFile: file of Out;
  OutData: Out;

          {$I B:BINTABLE.PAS}              {Listings of these   }
          {$I B:MIXTABLE.PAS}              {include files follow}
          {$I b:negLLike.pas}              {the listing of the  }
          {$I b:newp.pas    }              {main program        }

BEGIN
  assign(DataFile,'B:95707516.DAT'); reset(DataFile);
  assign(OutFile,'B:95707516.aic'); rewrite (OutFile);
  writeln('start,end');
```

```
  readln(start,finis);
  writeln(' ');
  seek(datafile,start);

repeat
  read(DataFile,Data);
  for indx:=1 to ssize do
   begin
   if (Data[indx]<0) or (Data[indx]>N) then
    begin
    writeln('Error in the range for the data in record',
     FilePos(Datafile)-1);
    goto DataError;
    end;
  end;

for indx:= 1 to ssize do
 begin
 {Initial estimates}
     mixprop:=indx/ssize;
     p0:=0.0;
     for jndx:=1 to indx do p0:=p0+Data[jndx];
     p0:=p0/N/indx;
     p1:=0.0;

     for jndx:=indx+1 to ssize do p1:=p1+Data[jndx];
     if ssize=indx then
       p1:=0.5
       else
       p1:=p1/N/(ssize-indx);

  repeat
    oldp0:=p0; oldp1:=p1;
    BinTable(p0,N,cf0,f0);
    BinTable(p1,N,cf1,f1);
    MixTable(mixprop,0,N,cf0,f0,cf1,f1,cm,m);
    Newp(f0,f1,m,Data,ssize,N,p0,p1);
  until (abs(p0-oldp0)<er) and (abs(p1-oldp1)<er);

  negllike(m,Data,ssize,newll);
  aic:=2*newll+4;
  if indx=ssize then aic:=aic-2;
  if indx=1 then minaic:=aic;
  if aic <= minaic then
    begin
    minaic:=aic;
    with OutData do
       begin
       prob0:=p0;
       prob1:=p1;
       mixprob:=mixprop;
       DataNum:=FilePos(DataFile)-1;
```

```
    end;
  end;

end;
  {write(OutFile,OutData);}
writeln(lst,OutData.prob0:7:3,OutData.prob1:7:3,OutData.mixprob:7
     :3,16*Outdata.mixprob:5:1,OutData.DataNum:6);
writeln(lst,' ');(form feed)
writeln(OutData.prob0:7:3,OutData.prob1:7:3,OutData.mixprob:7:3,
     16*Outdata.mixprob:5:1,OutData.DataNum:6);
DataError: ;
{until eof(DataFile);}
until FilePos(Datafile)=finis;

flush(OutFile);
close(OutFile);
close(DataFile);

END.

procedure BinTable(p: Real; N: integer;
             var CumProbFunc, ProbFunc: ProbTable);
{Procedure to compute the cumulative probability function and the
     probability function of a Binomial distribution.}

Var
  indx: integer;
  lnprob: array[0..20] of real;
  prob, q: real;

begin
  q:=1-p;
  prob:=1;

  if p>=1.0 then
    begin
    for indx:= 0 to n-1 do
      begin
      CumProbFunc[indx]:=0.0;
      ProbFunc[N]:=1.0;
      end;

  if p<=0.0 then
    begin
    for indx:= 1 to no do
      begin
      CumProbFunc[indx]:=1.0;
      ProbFunc[indx]:=0.0;
      end;
  CumProbFunc[0]:=;/0;
  ProbFunc[0]:=1.0;
  end;
```

```
  if (p<0.0)and(p<1.0) then
  begin
  lnprob[0]:=N*ln(q);
  for indx:=1 to N do
    begin
    lnprob[indx]:=lnprob[indx-1]+n(p)-ln(q)+ln(N-indx+1)-
      ln(indx);
    end;

  for indx:=0 to N do Probfunc[indx]:=exp(lnprob[indx]);

  for indx:=0 to No do CumProbFunc[indx]:=0.0;
  CumProbFunc[0]:=ProbFunc[0]:
  for indx:=1 to No do CumProbFunc[indx]:=CumProbFunc[indx-1]+ProbFunc[indx[;

  CumProbFunc[N]:-1.0;
  end;
end:

procedure MixTable (mixprop:real; LowRange, UpRange:integer;
          CumProbfunc0, ProbFunc0, CumProbFunc1, ProbFunc1:
          ProbTable; var MixCumProbFunc, MixProbFunc: Prob:Table);
{Computes the cumulative probability function and the probability function
     of a mixture.}

Var

  indx: integer;

begin

for indix:=LowRange to UpRange do
begin
  MixProbFunc[indx]:=mixprob*PropFunc0[indx] + (1-mixprop)*ProbFunc1[indx];
  MixCumProbFunc[indx]:=mixprop*CumProbFunc0[indx] + (1-mixprop)*CumProb
    Func1[indx];
end;

end;

procedure negllike(density:ProbTable;data:Sample;SSize:integer;var NegLnLike:
                   real);

{Computes the negative of the log likelihood for the density function.}

Var
  indx: integer;

begin
  NegLnLike:=0.0;
  for indx:= 1 to SSize do NegLnLike:=NegLnLike - ln(density[data[indx]];
end;
```

```
procedure Newp(Density0, Densityl, MixDensity:ProbTable;
               Data: Sample;
               ssize. M: integer;
               var p0, pl: real);
{Uses formula from EM algorithm to compute new values of p0 and pl.}

var
  M0, Ml: real;
  indx: integer;

  begin
  p0:=0; Pl:=0;
  M0:=0; Ml:=0:
  for indx:=1 to ssize do
    begin
    M0:=M0+Density0[Data[indx]]/MixDensity[Data[indx]];
    Ml:=Ml+Densityl[Data[indx]]/MixDensity[Data[indx]];
    end;
  for indx:=1 to ssize do
    begin
  p0:p0+Data[indx]*Desnity0[Data[indx]]/MixDensity[Data[indx]];
  pl:pl+Data[indx]*Desnityl[Data[indx]]/MixDensity[Data[indx]];
    end;

  p0:+p0/N/M0;
  pl:+pl/N/Ml;
  end;
```

Note 1: Variables of type ProbTable that begin with the letter "c" are cumulative probability functions and are not needed for the present analysis.

Note 2: In procedure BINTABLE.PAS, the binomial probabilities are computed using logarithms, because computing them in the usual way causes an overflow error due to the fact that p_0 or p_1 could be very close to one or zero.

APPENDIX B

GRAPHS OF MIXTURES OF BINOMIALS

This appendix contains graphs of the probability functions.

$$\text{MixProp}\cdot\text{bin}(x;p_o,N) + (1-\text{MixProb})\cdot\text{bin}(x;p_1,N)$$

for the following four cases:

(1) $p_o=0.95$, $P_1=0.70$, MixProp=0.35
(2) $p_o=0.95$, $P_1=0.70$, MixProp=0.75
(3) $p_o=0.95$, $P_1=0.80$, MixProp=0.29
(4) $p_o=0.90$, MixProp=1.0

and where

$$\text{bin}(x;p,N) = (N!/[x!(N-x)!])p^x(1-p)^{n-x} \text{ for } x=0,\ldots,N$$

is the binomial probability function.

p0 = 0.95, p1 = 0.8, mixprop = 0.29
p0 = 0.95, p1 = 0.7, mixprop = 0.35
p0 = 0.9, mixprop = 1.0
p0 = 0.95, p1 = 0.7, mixprop = 0.75

INITIAL DISTRIBUTION

Copies		CENTER DISTRIBUTION	
		Copies	Code
2	ONR Arlington		
	1 431M (P. Clark)		
	1 431N	1	2759
		1	28
4	NAVSEA	2	2803
	1 SEA 05M (A. Kaznoff)	1	2809
	1 SEA 05M1 (S. Rogers)	1	284
	1 SEA 05R15	1	2841
	1 SEA 05R26 (J. DeCorpo. Gagorik)	1	2841 (Laster)
		1	2841 (Chasan)
1	NRL	10	2841 (Houghton)
	(Code 6120, R. Brady)	10	2844 (Castelli)
		1	2844
1	NAVSSES		522.2
	(Code 053C, F. Boyle)		5231
12	DTIC		

END

11-86

DTIC